Introduction to Reinforcement Learning

Lecture 4. Policy Gradient Methods

Sungjoon Choi, Korea University

Content

- Proving Policy Gradient Theorem
- Trust Region Policy Optimization (**TRPO**)
- Proximal Policy Optimization (**PPO**)
- Generalized Advantage Estimation (**GAE**)
- Soft Actor-Critic (**SAC**)

ROBOT INTELLIGENCE LAB

Policy Gradient Theorem

Policy Optimization

- Policy gradient methods cast reinforcement learning into an optimization problem.

Policy Optimization

ROBOT INTELLIGENCE LAB

- Policy gradient methods cast reinforcement learning into an optimization problem.

- Find θ that maximizes the return:

$$\eta(\pi_\theta) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t \,|\, \pi_\theta\right]$$

Policy Optimization

ROBOT INTELLIGENCE LAB

- Policy gradient methods cast reinforcement learning into an optimization problem.

- Find θ that maximizes the return:

$$\eta(\pi_\theta) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t \,|\, \pi_\theta\right]$$

- We update the parameters of the **policy** function by computing the **gradient** of the parameters of the objective function:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \eta(\pi_\theta)$$

How to compute the gradients

ROBOT INTELLIGENCE LAB

$$\nabla_\theta \eta(\pi_\theta) = \nabla_\theta \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t \,|\, \pi_\theta\right]$$

- Policy Gradient Theorem:

$$\nabla_\theta \eta(\pi_\theta) = \frac{1}{(1-\gamma)} \sum_s \rho_{\pi_\theta} \sum_a \nabla_\theta \pi_\theta(a \,|\, s) Q^{\pi_\theta}(s, a)$$

$$\nabla_\theta \eta(\pi_\theta) \approx \nabla_\theta \log \pi_\theta(a_t \,|\, s_t) Q_{\pi_\theta}(s_t, a_t)$$

- Note that we only require the gradient of $\pi_\theta(\,\cdot\,)$ not $Q^{\pi_\theta}(\,\cdot\,)$!

R. Sutton et al, "Policy Gradient Methods for Reinforcement Learning with Function Approximation", NeurIPS, 2000

State Visitation

- Stationary distribution of the state given $\pi_\theta(\cdot)$

$$\rho_{\pi_\theta}(s) = (1-\gamma)\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t \mathbb{I}_{(St=s)}\right] = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P_{\pi_\theta}(S_t = s)$$

State Visitation

- Stationary distribution of the state given $\pi_\theta(\cdot)$

$$\rho_{\pi_\theta}(s) = (1-\gamma)\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t \mathbb{I}_{(St=s)}\right] = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P_{\pi_\theta}(S_t = s)$$

- $\rho_{\pi\theta}(s)$ is a probability mass function

$$\sum_s \rho_{\pi_\theta}(s) = (1-\gamma)\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t \sum_s \mathbb{I}_{(S_t=s)}\right] = (1-\gamma)\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t\right] = (1-\gamma)\frac{1}{1-\gamma} = 1$$

Proof of Policy Gradient (1/10)

ROBOT INTELLIGENCE LAB

- Return $\eta(\pi_\theta)$ of a policy $\pi_\theta(\,\cdot\,)$ and its gradient:

$$\eta(\pi_\theta) = \sum_s d(s) V_{\pi_\theta}(s)$$

$$\nabla_\theta \eta(\pi_\theta) = \sum_s d(s) \nabla_\theta V_{\pi_\theta}(s)$$

Proof of Policy Gradient (1/10)

- Return $\eta(\pi_\theta)$ of a policy $\pi_\theta(\cdot)$ and its gradient:

$$\eta(\pi_\theta) = \sum_s d(s) V_{\pi_\theta}(s)$$

$$\nabla_\theta \eta(\pi_\theta) = \sum_s d(s) \nabla_\theta V_{\pi_\theta}(s)$$

- Let's select an arbitrary state, say s_1, and compute $\nabla_\theta V_{\pi_\theta}(s_1)$:

$$\nabla_\theta V_{\pi_\theta}(s_1) = \nabla_\theta \sum_a \pi_\theta(a \mid s_1) Q_{\pi_\theta}(s_1, a)$$

$$\nabla_\theta V_{\pi_\theta}(s_1) = \sum_a \nabla_\theta \pi_\theta(a \mid s_1) Q_{\pi_\theta}(s_1, a) + \pi_\theta(a \mid s_1) \nabla_\theta Q_{\pi_\theta}(s_1, a)$$

Proof of Policy Gradient (2/10)

$$\nabla_\theta V_{\pi_\theta}(s_1) = \sum_a \nabla_\theta \pi_\theta(a \mid s_1) Q_{\pi_\theta}(s_1, a) + \pi_\theta(a \mid s_1) \nabla_\theta Q_{\pi_\theta}(s_1, a)$$

Proof of Policy Gradient (2/10)

ROBOT INTELLIGENCE LAB

$$\nabla_\theta V_{\pi_\theta}(s_1) = \sum_a \nabla_\theta \pi_\theta(a \mid s_1) Q_{\pi_\theta}(s_1, a) + \pi_\theta(a \mid s_1) \nabla_\theta Q_{\pi_\theta}(s_1, a)$$

$$\Downarrow$$

$$\nabla_\theta V_{\pi_\theta}(s_1) = \sum_a \nabla_\theta \pi_\theta(a \mid s_1) Q_{\pi_\theta}(s_1, a) + \pi_\theta(a \mid s_1) \nabla_\theta \left[r(s_1, a) + \gamma \sum_{s'} V_{\pi_\theta}(s') P(s' \mid s_1, a) \right]$$

Proof of Policy Gradient (2/10)

ROBOT INTELLIGENCE LAB

$$\nabla_\theta V_{\pi_\theta}(s_1) = \sum_a \nabla_\theta \pi_\theta(a \mid s_1) Q_{\pi_\theta}(s_1, a) + \pi_\theta(a \mid s_1) \nabla_\theta Q_{\pi_\theta}(s_1, a)$$

$$\Downarrow$$

$$\nabla_\theta V_{\pi_\theta}(s_1) = \sum_a \nabla_\theta \pi_\theta(a \mid s_1) Q_{\pi_\theta}(s_1, a) + \pi_\theta(a \mid s_1) \nabla_\theta \left[r(s_1, a) + \gamma \sum_{s'} V_{\pi_\theta}(s') P(s' \mid s_1, a) \right]$$

$$\Downarrow$$

$$\nabla_\theta V_{\pi_\theta}(s_1) = \sum_a \nabla_\theta \pi_\theta(a \mid s_1) Q_{\pi_\theta}(s_1, a) + \pi_\theta(a \mid s_1) \left[\gamma \sum_{s'} \nabla_\theta V_{\pi_\theta}(s') P(s' \mid s_1, a) \right]$$

Proof of Policy Gradient (3/10)

ROBOT INTELLIGENCE LAB

$$\nabla_\theta V_{\pi_\theta}(s_1) = \sum_a \nabla_\theta \pi_\theta(a \mid s_1) Q_{\pi_\theta}(s_1, a) + \pi_\theta(a \mid s_1) \left[\gamma \sum_{s'} \nabla_\theta V_{\pi_\theta}(s') P(s' \mid s_1, a) \right]$$

$$\nabla_\theta V_{\pi_\theta}(s') = \sum_{a'} \nabla_\theta \pi_\theta(a' \mid s') Q_{\pi_\theta}(s', a') + \pi_\theta(a' \mid s') \left[\gamma \sum_{s''} \nabla_\theta V_{\pi_\theta}(s'') P(s'' \mid s', a') \right]$$

- By plugging in:

$$\nabla_\theta V_{\pi_\theta}(s_1) = \sum_a \nabla_\theta \pi_\theta(a \mid s_1) Q_{\pi_\theta}(s_1, a) + \pi_\theta(a \mid s_1) \left[\gamma \sum_{s'} \left[\sum_{a'} \nabla_\theta \pi_\theta(a' \mid s') Q_{\pi_\theta}(s', a') + \pi_\theta(a' \mid s') \left[\gamma \sum_{s''} \nabla_\theta V_{\pi_\theta}(s'') P(s'' \mid s', a') \right] \right] P(s' \mid s_1, a) \right]$$

Proof of Policy Gradient (4/10)

$$\nabla_\theta V_{\pi_\theta}(s_1) = \sum_a \left[\nabla_\theta \pi_\theta(a|s_1) Q_{\pi_\theta}(s_1, a) + \pi_\theta(a|s_1) \left[\gamma \sum_{s'} \left[\sum_{a'} \nabla_\theta \pi_\theta(a'|s') Q_{\pi_\theta}(s', a') + \pi_\theta(a'|s') \left[\gamma \sum_{s''} \nabla_\theta V_{\pi_\theta}(s'') P(s''|s', a') \right] \right] P(s'|s_1, a) \right] \right]$$

- $\nabla_\theta V_{\pi_\theta}(s_1)$ contains three terms

$$\sum_a \nabla_\theta \pi_\theta(a|s_1) Q_{\pi_\theta}(s_1, a)$$

$$+ \sum_a \pi_\theta(a|s_1) \left[\gamma \sum_{s'} \left[\sum_{a'} \nabla_\theta \pi_\theta(a'|s') Q_{\pi_\theta}(s', a') \right] P(s'|s_1, a) \right]$$

$$+ \sum_a \pi_\theta(a|s_1) \left[\gamma \sum_{s'} \left[\sum_{a'} \pi_\theta(a'|s') \left[\gamma \sum_{s''} \nabla_\theta V_{\pi_\theta}(s'') P(s''|s', a') \right] \right] P(s'|s_1, a) \right]$$

Proof of Policy Gradient (5/10)

$\nabla_\theta V_{\pi_\theta}(s_1)$ contains three terms

$$
\begin{aligned}
&\sum_a \nabla_\theta \pi_\theta(a|s_1) Q_{\pi_\theta}(s_1, a) \\
&+ \sum_a \pi_\theta(a|s_1)\left[\gamma \sum_{s'}\left[\sum_{a'} \nabla_\theta \pi_\theta(a'|s') Q_{\pi_\theta}(s', a')\right] P(s'|s_1, a)\right] \\
&+ \sum_a \pi_\theta(a|s_1)\left[\gamma \sum_{s'}\left[\sum_{a'} \pi_\theta(a'|s')\left[\gamma \sum_{s''} \nabla_\theta V_{\pi_\theta}(s'') P(s''|s', a')\right]\right] P(s'|s_1, a)\right]
\end{aligned}
$$

- If we focus on the **first** term

$$\sum_a \nabla_\theta \pi_\theta(a|s_1) Q_{\pi_\theta}(s_1, a)$$

$P_{\pi_\theta}(S_0 = s' | S_0 = s_1)$ is nonzero only when $s' = s_1$

$$\sum_s \sum_a \nabla_\theta \pi_\theta(a|s_1) Q_{\pi_\theta}(s, a) P_{\pi_\theta}(S_0 = s | S_0 = s_1)$$

Proof of Policy Gradient (6/10)

$\nabla_\theta V_{\pi_\theta}(s_1)$ contains three terms

$$\sum_a \nabla_\theta \pi_\theta(a|s_1) Q_{\pi_\theta}(s_1, a)$$
$$+ \sum_a \pi_\theta(a|s_1)\left[\gamma \sum_{s'} \left[\sum_{a'} \nabla_\theta \pi_\theta(a'|s') Q_{\pi_\theta}(s', a')\right] P(s'|s_1, a)\right]$$
$$+ \sum_a \pi_\theta(a|s_1)\left[\gamma \sum_{s'} \left[\sum_{a'} \pi_\theta(a'|s') \left[\gamma \sum_{s''} \nabla_\theta V_{\pi_\theta}(s'') P(s''|s', a')\right]\right] P(s'|s_1, a)\right]$$

- If we focus on the **second** term

$$\sum_a \pi_\theta(a|s_1)\gamma \sum_{s'} \sum_{a'} \nabla_\theta \pi_\theta(a'|s') Q_{\pi_\theta}(s', a') P(s'|s_1, a)$$

Rearrange

$$\sum_{s'} \sum_{a'} \nabla_\theta \pi_\theta(a'|s') Q_{\pi_\theta}(s', a') \gamma \sum_a P(s'|s_1, a) \pi_\theta(a|s_1)$$

State Transition Probability

$$\sum_{s'} \sum_{a'} \nabla_\theta \pi_\theta(a'|s') Q_{\pi_\theta}(s', a') \gamma P_{\pi_\theta}(S_1 = s'|S_0 = s_1)$$

Proof of Policy Gradient (7/10)

$\nabla_\theta V_{\pi_\theta}(s_1)$ contains three terms

$$\sum_a \nabla_\theta \pi_\theta(a|s_1) Q_{\pi_\theta}(s_1, a)$$
$$+ \sum_a \pi_\theta(a|s_1)\left[\gamma \sum_{s'}\left[\sum_{a'} \nabla_\theta \pi_\theta(a'|s') Q_{\pi_\theta}(s', a')\right] P(s'|s_1, a)\right]$$
$$+ \sum_a \pi_\theta(a|s_1)\left[\gamma \sum_{s'}\left[\sum_{a'} \pi_\theta(a'|s')\left[\gamma \sum_{s''} \nabla_\theta V_{\pi_\theta}(s'') P(s''|s', a')\right]\right] P(s'|s_1, a)\right]$$

- If we focus on the **third** term

$$\sum_a \pi_\theta(a|s_1)\gamma \sum_{s'}\sum_{a'} \pi_\theta(a'|s')\gamma \sum_{s''} \nabla_\theta V_{\pi_\theta}(s'') P(s''|s', a') P(s'|s_1, a)$$

Rearrange

$$\sum_{s''} \nabla_\theta V_{\pi_\theta}(s'')\gamma^2 \sum_a \sum_{s'} \sum_{a'} \pi_\theta(a|s_1)\pi_\theta(a'|s') P(s''|s', a') P(s'|s_1, a)$$

State Transition Probability

$$\sum_{s''} \nabla_\theta V_{\pi_\theta}(s'')\gamma^2 P_{\pi_\theta}(S_2 = s''|S_0 = s_1)$$

Proof of Policy Gradient (8/10)

ROBOT INTELLIGENCE LAB

- Substituting the first, second and third terms:

$$\nabla_\theta V_{\pi_\theta}(s_1) = \sum_s \sum_a \nabla_\theta \pi_\theta(a|s_1) Q_{\pi_\theta}(s,a) P_{\pi_\theta}(S_0 = s | S_0 = s_1) + \sum_{s'} \sum_{a'} \nabla_\theta \pi_\theta(a'|s') Q_{\pi_\theta}(s',a') \gamma P_{\pi_\theta}(S_1 = s' | S_0 = s_1) + \sum_{s''} \nabla_\theta V_{\pi_\theta}(s'') \gamma^2 P_{\pi_\theta}(S_2 = s'' | S_0 = s_1)$$

Mathematical Induction

$$\nabla_\theta V_{\pi_\theta}(s_1) = \sum_s \sum_a \nabla_\theta \pi_\theta(a|s) Q_{\pi_\theta}(s,a) \left(P_{\pi_\theta}(S_0 = s | S_0 = s_1) + \gamma P_{\pi_\theta}(S_1 = s | S_0 = s_1) + \gamma^2 P_{\pi_\theta}(S_2 = s | S_0 = s_1) + \gamma^3 P_{\pi_\theta}(S_3 = s | S_0 = s_1) + \cdots \right)$$

$$\nabla_\theta V_{\pi_\theta}(s_1) = \sum_s \sum_a \nabla_\theta \pi_\theta(a|s) Q_{\pi_\theta}(s,a) \sum_{t=0}^{\infty} \gamma^t P_{\pi_\theta}(S_t = s | S_0 = s_1)$$

Proof of Policy Gradient (9/10)

- Plugging in $\nabla_\theta V_{\pi_\theta}(s_1) = \sum_s \sum_a \nabla_\theta \pi_\theta(a \mid s) Q_{\pi_\theta}(s,a) \sum_{t=0}^{\infty} \gamma^t P_{\pi_\theta}(S_t = s \mid S_0 = s_1)$ to $\nabla_\theta \eta(\pi_\theta) = \sum_s d(s) \nabla_\theta V_{\pi_\theta}(s)$

Proof of Policy Gradient (9/10)

ROBOT INTELLIGENCE LAB

- Plugging in $\nabla_\theta V_{\pi_\theta}(s_1) = \sum_s \sum_a \nabla_\theta \pi_\theta(a|s) Q_{\pi_\theta}(s,a) \sum_{t=0}^{\infty} \gamma^t P_{\pi_\theta}(S_t = s | S_0 = s_1)$ to $\nabla_\theta \eta(\pi_\theta) = \sum_s d(s) \nabla_\theta V_{\pi_\theta}(s)$

$$\nabla_\theta \eta(\pi_\theta) = \sum_s d(s) \sum_{s'} \sum_a \nabla_\theta \pi_\theta(a|s') Q_{\pi_\theta}(s',a) \sum_{t=0}^{\infty} \gamma^t P_{\pi_\theta}(S_t = s' | S_0 = s)$$

Proof of Policy Gradient (9/10)

ROBOT INTELLIGENCE LAB

- Plugging in $\nabla_\theta V_{\pi_\theta}(s_1) = \sum_s \sum_a \nabla_\theta \pi_\theta(a|s) Q_{\pi_\theta}(s,a) \sum_{t=0}^{\infty} \gamma^t P_{\pi_\theta}(S_t = s | S_0 = s_1)$ to $\nabla_\theta \eta(\pi_\theta) = \sum_s d(s) \nabla_\theta V_{\pi_\theta}(s)$

$$\nabla_\theta \eta(\pi_\theta) = \sum_s d(s) \sum_{s'} \sum_a \nabla_\theta \pi_\theta(a|s') Q_{\pi_\theta}(s',a) \sum_{t=0}^{\infty} \gamma^t P_{\pi_\theta}(S_t = s' | S_0 = s)$$

Rearrange

$$= \sum_{s'} \sum_a \nabla_\theta \pi_\theta(a|s') Q_{\pi_\theta}(s',a) \sum_{t=0}^{\infty} \gamma^t \sum_s P_{\pi_\theta}(S_t = s' | S_0 = s) d(s)$$

Proof of Policy Gradient (9/10)

ROBOT INTELLIGENCE LAB

- Plugging in $\nabla_\theta V_{\pi_\theta}(s_1) = \sum_s \sum_a \nabla_\theta \pi_\theta(a \mid s) Q_{\pi_\theta}(s,a) \sum_{t=0}^{\infty} \gamma^t P_{\pi_\theta}(S_t = s \mid S_0 = s_1)$ to $\nabla_\theta \eta(\pi_\theta) = \sum_s d(s) \nabla_\theta V_{\pi_\theta}(s)$

$$\nabla_\theta \eta(\pi_\theta) = \sum_s d(s) \sum_{s'} \sum_a \nabla_\theta \pi_\theta(a \mid s') Q_{\pi_\theta}(s',a) \sum_{t=0}^{\infty} \gamma^t P_{\pi_\theta}(S_t = s' \mid S_0 = s)$$

$$= \sum_{s'} \sum_a \nabla_\theta \pi_\theta(a \mid s') Q_{\pi_\theta}(s',a) \sum_{t=0}^{\infty} \gamma^t \sum_s P_{\pi_\theta}(S_t = s' \mid S_0 = s) d(s)$$

By definition

$$= \sum_{s'} \sum_a \nabla_\theta \pi_\theta(a \mid s') Q_{\pi_\theta}(s',a) \sum_{t=0}^{\infty} \gamma^t P_{\pi_\theta}(S_t = s')$$

Proof of Policy Gradient (9/10)

ROBOT INTELLIGENCE LAB

- Plugging in $\nabla_\theta V_{\pi_\theta}(s_1) = \sum_s \sum_a \nabla_\theta \pi_\theta(a|s) Q_{\pi_\theta}(s,a) \sum_{t=0}^{\infty} \gamma^t P_{\pi_\theta}(S_t = s | S_0 = s_1)$ to $\nabla_\theta \eta(\pi_\theta) = \sum_s d(s) \nabla_\theta V_{\pi_\theta}(s)$

$$\nabla_\theta \eta(\pi_\theta) = \sum_s d(s) \sum_{s'} \sum_a \nabla_\theta \pi_\theta(a|s') Q_{\pi_\theta}(s',a) \sum_{t=0}^{\infty} \gamma^t P_{\pi_\theta}(S_t = s' | S_0 = s)$$

$$= \sum_{s'} \sum_a \nabla_\theta \pi_\theta(a|s') Q_{\pi_\theta}(s',a) \sum_{t=0}^{\infty} \gamma^t \sum_s P_{\pi_\theta}(S_t = s' | S_0 = s) d(s)$$

$$= \sum_{s'} \sum_a \nabla_\theta \pi_\theta(a|s') Q_{\pi_\theta}(s',a) \sum_{t=0}^{\infty} \gamma^t P_{\pi_\theta}(S_t = s')$$

$$\rho_{\pi_\theta}(s) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t P_{\pi_\theta}(S_t = s)$$

$$= \frac{1}{1-\gamma} \sum_{s'} \sum_a \nabla_\theta \pi_\theta(a|s') Q_{\pi_\theta}(s',a) \rho_{\pi_\theta}(s')$$

Proof of Policy Gradient (10/10)

ROBOT INTELLIGENCE LAB

$$\nabla_\theta \eta(\pi_\theta) = \frac{1}{1-\gamma} \sum_{s'} \sum_{a} \nabla_\theta \pi_\theta(a \mid s') Q_{\pi_\theta}(s', a) \rho_{\pi_\theta}(s')$$

$$\propto \mathbb{E}_{s \sim \rho_{\pi_\theta}} \left[\sum_{a} \nabla_\theta \pi_\theta(a \mid s) Q_{\pi_\theta}(s, a) \right]$$

- Note that the states should be sampled from the distribution induced from the current policy (i.e., $s \sim \rho_{\pi_\theta}(s)$)
- This makes policy gradient methods **on-policy**.

Log Ratio Trick

ROBOT INTELLIGENCE LAB

$$\nabla_\theta \eta(\pi_\theta) = \frac{1}{1-\gamma} \sum_{s'} \sum_{a} \nabla_\theta \pi_\theta(a \mid s') Q_{\pi_\theta}(s', a) \rho_{\pi_\theta}(s')$$

- However, we should summate over all possible states and actions.

Log Ratio Trick

ROBOT INTELLIGENCE LAB

$$\nabla_\theta \eta(\pi_\theta) = \frac{1}{1-\gamma} \sum_{s'} \sum_{a} \nabla_\theta \pi_\theta(a \mid s') Q_{\pi_\theta}(s', a) \rho_{\pi_\theta}(s')$$

- However, we should summate over all possible states and actions.

- We can use the log ratio trick to overcome this issue.

$$\nabla_\theta \mathbb{E}\left[f(x)\right] = \sum_x f(x) \nabla_\theta p_\theta(x) = \sum_x f(x) p_\theta(x) \frac{\nabla_\theta p_\theta(x)}{p_\theta(x)} = \sum_x f(x) p_\theta(x) \nabla_\theta \log p_\theta(x) = \mathbb{E}\left[f(x) \nabla_\theta \log p_\theta(x)\right]$$

- To summarize

$$\nabla_\theta \mathbb{E}\left[f(x)\right] = \mathbb{E}\left[f(x) \nabla_\theta \log p_\theta(x)\right]$$

Log Ratio Trick

$$\nabla_\theta \eta(\pi_\theta) = \frac{1}{1-\gamma}\sum_{s'}\sum_{a}\nabla_\theta \pi_\theta(a\,|\,s')Q_{\pi_\theta}(s',a)\rho_{\pi_\theta}(s')$$

$$= \frac{1}{1-\gamma}\sum_{s'}\sum_{a}\pi_\theta(a\,|\,s')\frac{\nabla_\theta \pi_\theta(a\,|\,s')}{\pi_\theta(a\,|\,s')}Q_{\pi_\theta}(s',a)\rho_{\pi_\theta}(s')$$

$$= \frac{1}{1-\gamma}\sum_{s'}\sum_{a}\pi_\theta(a\,|\,s')\,\nabla_\theta \log \pi_\theta(a\,|\,s')Q_{\pi_\theta}(s',a)\rho_{\pi_\theta}(s')$$

$$= \frac{1}{1-\gamma}\mathbb{E}_{a\sim\pi_\theta, s\sim\rho_\pi(S)}\left[\nabla_\theta \log \pi_\theta(a\,|\,s')Q_{\pi_\theta}(s',a)\right]$$

$$\approx \nabla_\theta \log \pi_\theta(a_t\,|\,s_t)Q_{\pi_\theta}(s_t,a_t)$$

- To summarize

$$\nabla_\theta \eta(\pi_\theta) \approx \nabla_\theta \log \pi_\theta(a_t\,|\,s_t)Q_{\pi_\theta}(s_t,a_t)$$

Trust Region Policy Optimization (TRPO)

"Trust Region Policy Optimization", 2015

PG as an optimization problem

ROBOT INTELLIGENCE LAB

- Policy-based reinforcement learning is an optimization problem.

- The goal is to find θ that maximizes

$$\eta(\pi_\theta) = \mathbb{E}_{s,a}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t)\right]$$

where $s_0 \sim \rho_0(s)$, $a_t \sim \pi(a_t | s_t)$, $s_{t+1} \sim P(s_{t+1} | s_t, a_t)$.

PG as an optimization problem

- Policy-based reinforcement learning is an optimization problem.

- The goal is to find θ that maximizes

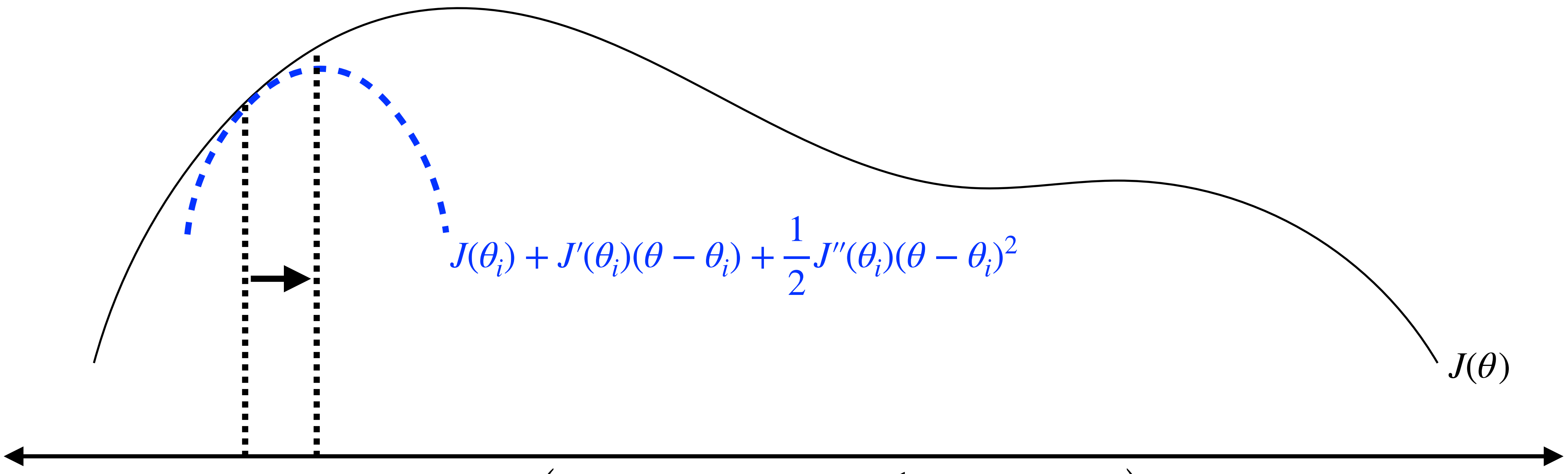
$$\eta(\pi_\theta) = \mathbb{E}_{s,a}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t)\right]$$

where $s_0 \sim \rho_0(s)$, $a_t \sim \pi(a_t | s_t)$, $s_{t+1} \sim P(s_{t+1} | s_t, a_t)$.

- We can use optimization techniques:
  - Minorization maximization
  - Conjugate gradient descent

Newton Method

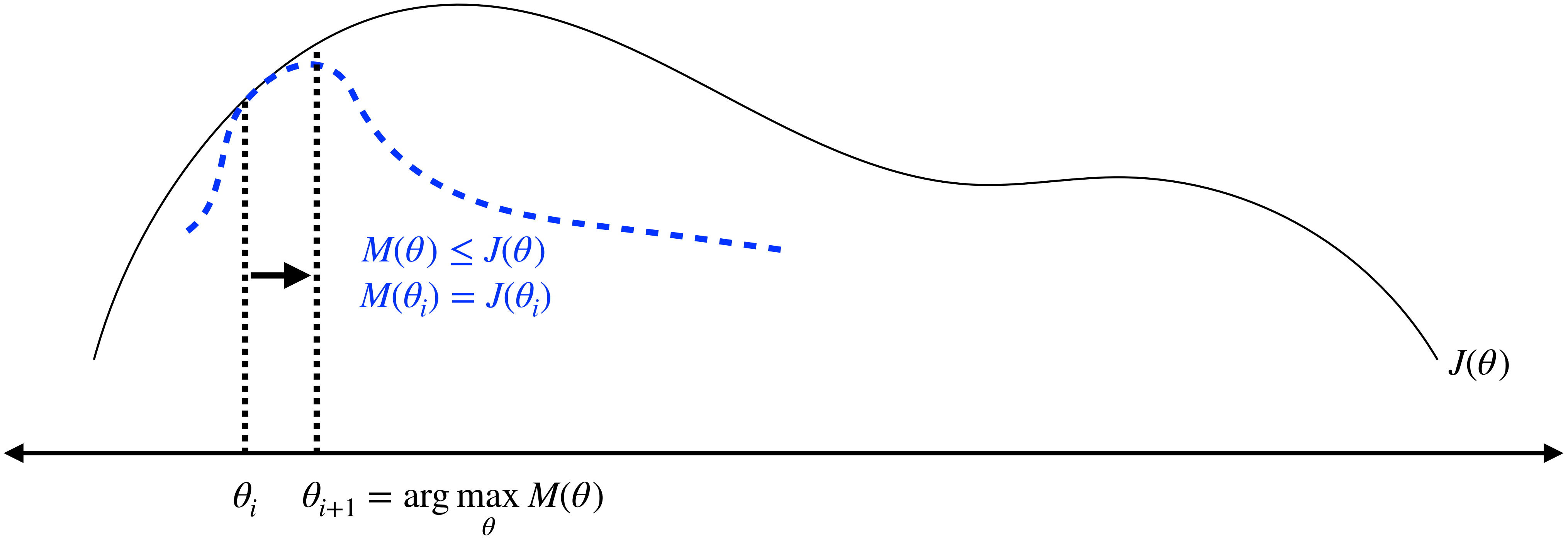
$$\max_{\theta} J(\theta)$$



$$\theta_{i+1} = \arg\max_{\theta} \left(J(\theta_i) + J'(\theta_i)(\theta - \theta_i) + \frac{1}{2}J''(\theta_i)(\theta - \theta_i)^2 \right)$$

Minorization Maximization

$$\max_{\theta} J(\theta)$$



Preliminaries

- The goal of reinforcement learning is to find π_θ that maximizes the expected return:

$$\eta(\pi_\theta) = \mathbb{E}_{s,a}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t)\right]$$

where $s_0 \sim \rho_0(s)$, $a_t \sim \pi(a_t | s_t)$, $s_{t+1} \sim P(s_{t+1} | s_t, a_t)$.

- Basic definitions of Markov decision processes

$$Q_\pi(s_t, a_t) = \mathbb{E}_{s,a}\left[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})\right]$$

$$V_\pi(s_t) = \mathbb{E}_{s,a}\left[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})\right]$$

$$A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s)$$

Useful Identity

ROBOT INTELLIGENCE LAB

- The improvement of the expected return:

$$\eta(\pi') = \eta(\pi) + \boxed{\mathbb{E}_{s,a\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t A_\pi(s_t, a_t)\right]}$$

Improvement of π' over π

- Let $\rho_\pi(s)$ be the (unnormalized) discounted visitation frequencies

$$\rho_\pi(s) = P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s) + \cdots$$

- Then the return improvement can be written as

$$\eta(\pi') = \eta(\pi) + \sum_s \rho_{\pi'}(s) \sum_a \pi'(a\,|\,s) A_\pi(s, a).$$

Performance Improvement

ROBOT INTELLIGENCE LAB

- The return improvement

$$\eta(\pi') = \eta(\pi) + \sum_{s} \rho_{\pi'}(s) \sum_{a} \pi'(a \mid s) A_{\pi}(s, a)$$

- Then, the **policy improvement** step of policy iteration will increase the policy performance if the following is guaranteed:

$$\sum_{a} \pi'(a \mid s) A_{\pi}(s, a) \geq 0$$

- Hence, $\pi'(s) = \arg\max_{a} A_{\pi}(s, a)$ will improve the policy there is at least one state-action pair with a positive value per each state s.
- However, due to the approximation of $A_{\pi}(s, a)$, it is not always guaranteed.

Performance Improvement

ROBOT INTELLIGENCE LAB

- The following return improvement is not practical due to $\rho_{\pi'}(s)$:

$$\eta(\pi') = \eta(\pi) + \sum_s \rho_{\pi'}(s) \sum_a \pi'(a \mid s) A_\pi(s, a)$$

- Why?

Performance Improvement

- The following return improvement is not practical due to $\rho_{\pi'}(s)$:

$$\eta(\pi') = \eta(\pi) + \sum_s \rho_{\pi'}(s) \sum_a \pi'(a|s) A_\pi(s,a)$$

- Why?

- The following local approximation, $\rho_{\pi'}(s) \Rightarrow \rho_\pi(s)$, is made:

$$L_\pi(\pi') \approx \eta(\pi')$$

$$L_\pi(\pi') = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \pi'(a|s) A_\pi(s,a)$$

Performance Improvement

- Local approximation:

$$L_{\pi}(\pi') = \eta(\pi) + \sum_{s} \rho_{\pi}(s) \sum_{a} \pi'(a \mid s) A_{\pi}(s, a)$$

- Hence the following can be used as a learning objective:

$$\mathbb{E}_{s_t \sim P, a_t \sim \pi'} \left[\sum_{t=0}^{\infty} \gamma^t A_{\pi}(s_t, a_t) \right]$$

- If we want to use the state-action pairs collected from the current policy π, we can use importance-sampling:

$$\mathbb{E}_{s_t \sim P, a_t \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \frac{\pi'(a_t \mid s_t)}{\pi(a_t \mid s_t)} A_{\pi}(s_t, a_t) \right]$$

$$= \mathbb{E}_{s_t \sim \rho_{\pi}, a_t \sim \pi} \left[\frac{\pi'(a_t \mid s_t)}{\pi(a_t \mid s_t)} A_{\pi}(s_t, a_t) \right]$$

Minorization for RL

ROBOT INTELLIGENCE LAB

$$L_\pi(\pi') = \mathbb{E}_{s\sim\rho_\pi, a\sim\pi}\left[\frac{\pi'(a_t|s_t)}{\pi(a_t|s_t)} A_\pi(s_t, a_t)\right]$$

- We can define the following **minorization** of $\eta(\pi)$ using KLD

$$M_\pi(\pi') = \eta(\pi) + L_\pi(\pi') - cD_{KL}^{max}(\pi, \pi')$$

where $D_{KL}^{max}(\pi, \pi') = \max_s D_{KL}\left(\pi(\cdot|s), \pi'(\cdot|s)\right)$

- Then the following properties hold:

$$M_\pi(\pi) = \eta(\pi)$$

$$M_\pi(\pi') \leq \eta(\pi')$$

Minorization for RL

- Now, we optimize

$$L_\pi(\pi') = \mathbb{E}_{s\sim\rho_\pi, a\sim\pi}\left[\frac{\pi'(a|s)}{\pi(a|s)}A_\pi(s,a)\right]$$

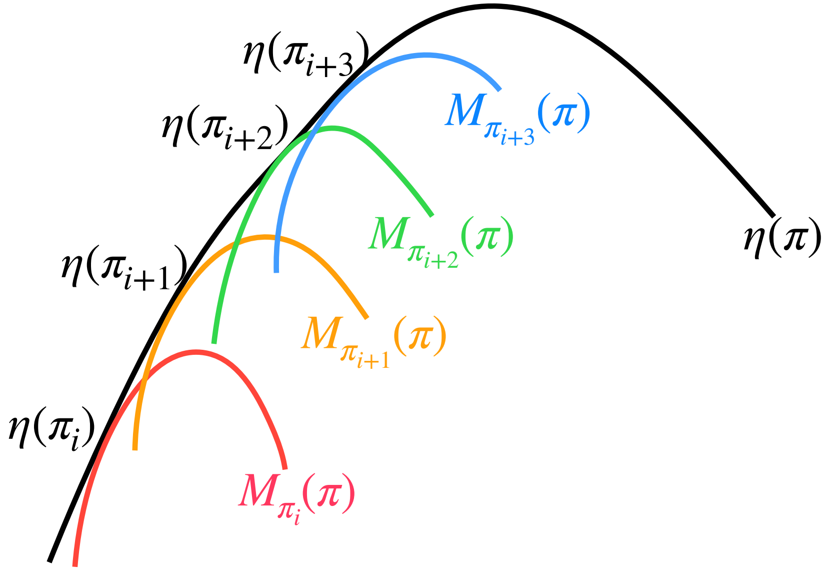
$$M_\pi(\pi') = \eta(\pi) + L_\pi(\pi') - cD_{KL}^{max}(\pi, \pi')$$

$$\max_{\theta_{i+1}} M_{\pi_{\theta_i}}(\pi_{\theta_{i+1}}) = \eta(\pi_{\theta_i}) + L_{\pi_{\theta_i}}(\pi_{\theta_{i+1}}) - cD_{KL}^{max}(\pi_{\theta_i}, \pi_{\theta_{i+1}})$$

- Lagrangian relaxation becomes

$$\max_{\theta_{i+1}} L_{\pi_{\theta_i}}(\pi_{\theta_{i+1}})$$

$$\text{subject to } D_{KL}^{max}(\pi_\theta, \pi_{\theta_{i+1}}) \leq \delta$$



Trust Region Policy Optimization

ROBOT INTELLIGENCE LAB

$$\max_{\theta_{i+1}} L_{\pi_{\theta_i}}(\pi_{\theta_{i+1}}) = \mathbb{E}_{s\sim\rho_{\pi_{\theta_i}}, a\sim\pi_{\theta_i}}\left[\frac{\pi_{\theta_{i+1}}(a\,|\,s)}{\pi_{\theta_i}(a\,|\,s)} A_{\pi_{\theta_i}}(s,a)\right]$$

$$\text{subject to } D_{KL}^{max}(\pi_\theta, \pi_{\theta_{i+1}}) \le \delta$$

- We approximate the KL divergence:

$$D_{KL}^{max}(\pi_\theta, \pi_{\theta_{i+1}}) = \boxed{\max_s} D_{KL}\left(\pi_{\theta_i}(\,\cdot\,|\,s), \pi_{\theta_{i+1}}(\,\cdot\,|\,s)\right)$$

$$\downarrow$$

$$D_{KL}^{\rho}(\pi_\theta, \pi_{\theta_{i+1}}) = \mathbb{E}_{s\sim\rho_{\pi_{\theta_i}}}\left[D_{KL}\left(\pi_{\theta_i}(\,\cdot\,|\,s), \pi_{\theta_{i+1}}(\,\cdot\,|\,s)\right)\right]$$

Trust Region Policy Optimization

$$\max_{\theta_{i+1}} L_{\pi_{\theta_i}}(\pi_{\theta_{i+1}}) = \mathbb{E}_{s\sim\rho_{\pi_{\theta_i}}, a\sim\pi_{\theta_i}}\left[\frac{\pi_{\theta_{i+1}}(a\,|\,s)}{\pi_{\theta_i}(a\,|\,s)} A_{\pi_{\theta_i}}(s,a)\right]$$

subject to $D_{KL}^{\rho}(\pi_\theta, \pi_{\theta_{i+1}}) \leq \delta$

- In summary,
  - TRPO is a minorization maximization framework for RL.
  - Interpretation of the trust region method:
    1. Update policy distribution slowly
    2. Consider the geometry of the distribution space
  - There are two approximations: 1) $\mathbb{E}_{s\sim\rho_{\pi'}} \Rightarrow \mathbb{E}_{s\sim\rho_{\pi}}$ and 2) $D_{KL}^{\max} \Rightarrow D_{KL}^{\rho}$

Trust Region Policy Optimization

ROBOT INTELLIGENCE LAB

- How do we estimate $A_{\pi_{\theta_i}}$?
- We estimate $Q_{\pi_{\theta_i}}(s, a)$ instead of $A_{\pi_{\theta_i}}(s, a)$:

$$A_{\pi_{\theta_i}}(s, a) = Q_{\pi_{\theta_i}}(s, a) - V_{\pi_{\theta_i}}(s)$$

- We use the Monte Carlo Estimate of Q:

$$Q_{\pi_{\theta_i}}(s_t, a_t) \approx G_t = \sum_{k=1} \gamma^k R_{t+1+k}$$

Trust Region Policy Optimization

ROBOT INTELLIGENCE LAB

$$\max_{\theta_{i+1}} L_{\pi_{\theta_i}}(\pi_{\theta_{i+1}}) = \mathbb{E}_{s\sim\rho_{\pi_{\theta_i}}, a\sim\pi_{\theta_i}}\left[\frac{\pi_{\theta_{i+1}}(a\,|\,s)}{\pi_{\theta_i}(a\,|\,s)} A_{\pi_{\theta_i}}(s,a)\right]$$

$$\text{subject to } D_{KL}^{\rho}(\pi_\theta, \pi_{\theta_{i+1}}) \le \delta$$

Linear approximation to the loss and
quadratic approximation to the constraint

$$\max_{\theta} \nabla_\theta L_{\theta_{old}}(\theta)\,|_{\theta=\theta_{old}} \cdot (\theta - \theta_{old})$$

$$\text{subject to } \frac{1}{2}(\theta_{old} - \theta)^T H(\theta_{old})(\theta_{old} - \theta) \le \delta$$

$$\text{where } H(\theta_{old})_{(i,j)} = \frac{\partial}{\partial\theta_i}\frac{\partial}{\partial\theta_j}\mathbb{E}_{s\sim\rho_\pi}\left[D_{KL}(\pi(\cdot\,|\,s,\theta_{old})\|\pi(\cdot\,|\,s,\theta))\right]|_{\theta=\theta_{old}}$$

Trust Region Policy Optimization

ROBOT INTELLIGENCE LAB

- The final TRPO objective becomes:

$$\max_{\theta} g(\theta_{old})^T(\theta - \theta_{old})$$

$$\text{subject to } \frac{1}{2}(\theta_{old} - \theta)^T H(\theta_{old})(\theta_{old} - \theta) \le \delta$$

$$\text{where } g(\theta_{old}) = \nabla_{\theta} L_{\theta_{old}}(\theta)\,|_{\theta=\theta_{old}} \text{ and}$$

$$H(\theta_{old})_{(i,j)} = \frac{\partial}{\partial\theta_i}\frac{\partial}{\partial\theta_j}\mathbb{E}_{s\sim\rho_\pi}\left[D_{KL}(\pi(\cdot\,|\,s,\theta_{old})\|\pi(\cdot\,|\,s,\theta))\right]|_{\theta=\theta_{old}}$$

- The update rule of the above problem is

$$\theta_{new} = \theta_{old} + \frac{1}{\lambda}H(\theta_{old})^{-1}g(\theta_{old})$$

Trust Region Policy Optimization

- The update rule of the above problem is

$$\theta_{new} = \theta_{old} + \frac{1}{\lambda} H(\theta_{old})^{-1} g(\theta_{old})$$

- However, the hessian matrix $H(\theta_{old}) \in \mathbb{R}^{n \times n}$ where n is the number of parameters and the computational complexity of the inverse becomes $O(n^3)$.

- Instead of computing H^{-1}, we solve the linear equation $Hx = g$ using a conjugate gradient method.

Proximal Policy Optimization (PPO)

"Proximal Policy Optimization Algorithms", 2017

Preliminaries

ROBOT INTELLIGENCE LAB

- Policy gradient method
  - The gradient estimate of the policy w.r.t. the return is

$$\hat{g} = \mathbb{E}_{s_t, a_t}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \hat{A}_t \right]$$

where $\hat{A}_t$ is an estimator of the advantage function.

- Trust region method
  - The TRPO objective is

$$\max_\theta \mathbb{E}\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} \hat{A}_t \right]$$

$$\text{s.t. } D_{KL}^{\rho}\left[\pi_{old}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t) \right] \leq \delta$$

Clipped Surrogate Objective

ROBOT INTELLIGENCE LAB

- The objective of the TRPO is:

$$L(\theta) = \mathbb{E}\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}\hat{A}_t\right] = \mathbb{E}\left[r_t(\theta)\hat{A}_t\right]$$

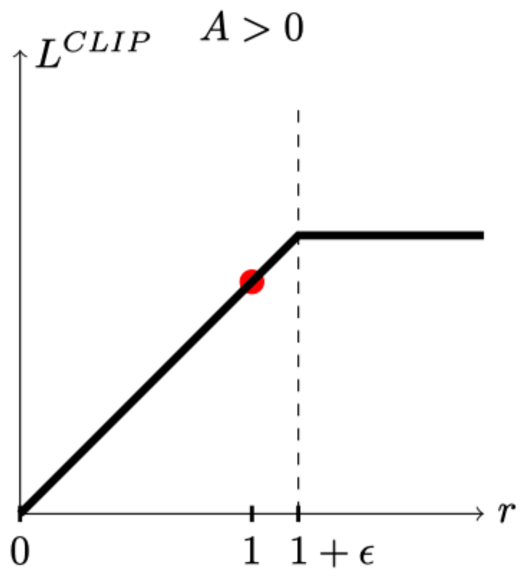
- The main objective of clipped surrogate is:

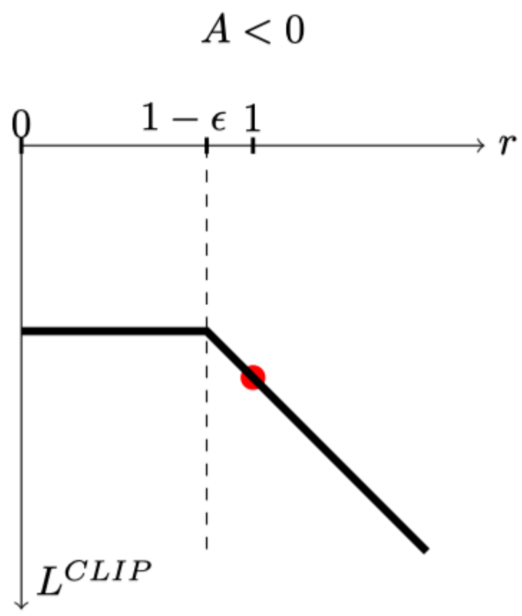
$$L^{\text{CLIP}}(\theta) = \mathbb{E}\left[\min\left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right]$$

  - The first term $r_t(\theta)\hat{A}_t$ is identical to the TRPO objective.
  - The second term clips the probability ratio $r_t(\theta)$, which removes the incentive for moving $r_t(\theta)$ outside of the interval $[1-\epsilon, 1+\epsilon]$.

Clipped Surrogate Objective

$$L(\theta) = \mathbb{E}\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t\right] = \mathbb{E}\left[r_t(\theta)\hat{A}_t\right] \text{ and } L^{\text{CLIP}}(\theta) = \mathbb{E}\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right]$$





- When $A_t > 0$, we have to worry about increasing $L(\theta)$ by increasing $r_t(\theta)$, and vice versa. Hence, we clip the objective when $r_t(\theta)$ exceeds $1+\epsilon$ when $A_t > 0$.

Proximal Policy Optimization (Adaptive KL Penalty)

ROBOT INTELLIGENCE LAB

- The TRPO objective is:

$$\max_\theta \mathbb{E}\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t\right] \text{ s.t. } D_{KL}^{\rho}\left[\pi_{old}(\cdot|s_t), \pi_\theta(\cdot|s_t)\right] \le \delta$$

- The unconstrained objective of TRPO is:

$$L(\theta) = \max_\theta \mathbb{E}\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t - \beta D_{KL}^{\rho}\left[\pi_{\theta_{old}}(\cdot|s_t), \pi_\theta(\cdot|s_t)\right]\right]$$

- The adaptive KL penalty method for PPO is to adaptively change β by checking $d = \mathbb{E}_t\left[D_{KL}[\pi_{\theta_{old}}, \pi_\theta]\right]$:
  - If $d < d_{targ}/1.5$, $\beta \leftarrow \beta/2$
  - If $d > d_{targ} \times 1.5$, $\beta \leftarrow \beta \times 2$

Generalized Advantage Estimation (GAE)

"HIGH-DIMENSIONAL CONTINUOUS CONTROL USING GENERALIZED ADVANTAGE ESTIMATION," 2018

Advantage Function Estimation

ROBOT INTELLIGENCE LAB

- Let V be an approximate value function. Then define

$$\delta_t^V = r_t + \gamma V(s_{t+1}) - V(s_t)$$

i.e., the TD residual of V with discount γ.

- Note that δ_t^V can be considered as an estimate of the advantage of the action a_t, i.e., $\hat{A}_t$. Now, let's define the following series:
  - $\hat{A}_t^{(1)} = \delta_t^V$
  - $\hat{A}_t^{(2)} = \delta_t^V + \gamma\delta_{t+1}^V$
  - $\hat{A}_t^{(3)} = \delta_t^V + \gamma\delta_{t+1}^V + \gamma^2\delta_{t+2}^V$
- Finally, we define the λ-exponentially-weighted average of $\hat{A}_t$:

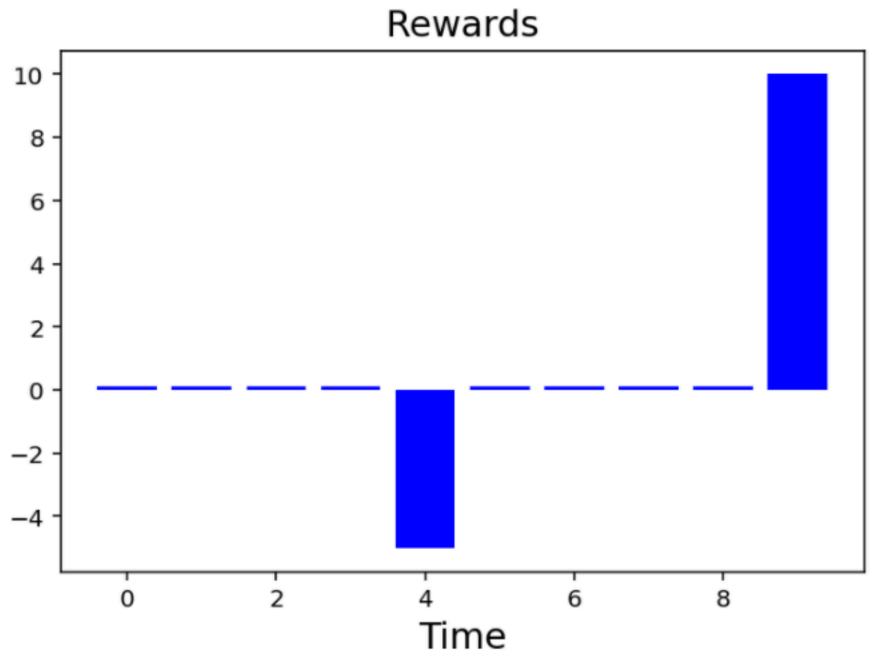
$$\hat{A}_t^{\text{GAE}(\gamma,\lambda)} = (1-\lambda)\left(\hat{A}_t^{(1)} + \lambda\hat{A}_t^{(2)} + \lambda^2\hat{A}_t^{(3)} + \cdots\right) = \sum_{t=0}^{\infty}(\gamma\lambda)^l\delta_{t+1}^V$$

Advantage Function Estimation

ROBOT INTELLIGENCE LAB

Rewards

```python
# Plot rewards
plt.bar(times,rewards,color='b')
plt.title("Rewards",fontsize=15)
plt.xlabel("Time",fontsize=15)
plt.show()
```

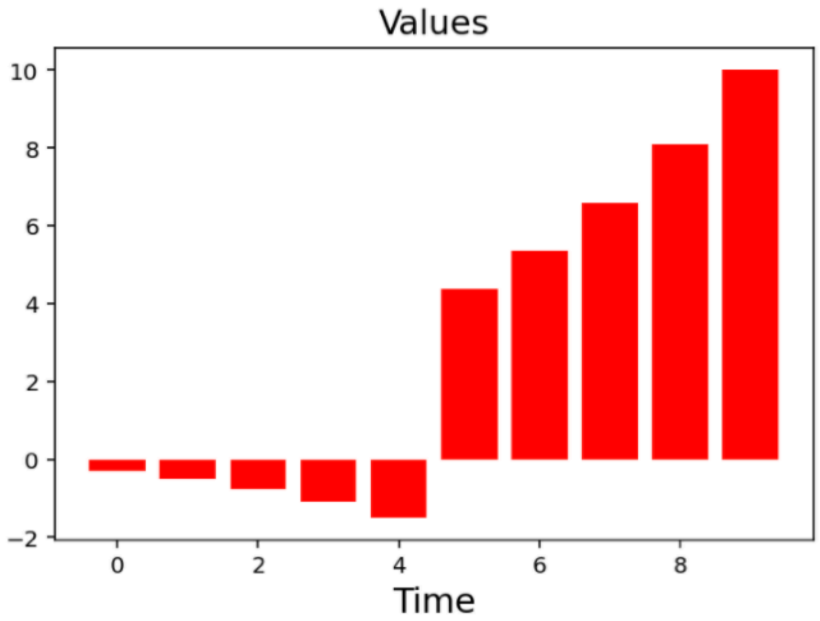


Advantage Function Estimation

Values

$V(s_t) = \sum_{l=0}^{\infty} \gamma^l r(s_{t+l})$ and $V(s_t) = r(s_t) + \gamma V(s_{t+1})$

```
values = np.zeros(L); values[L-1] = rewards[L-1]
for t in reversed(range(L-1)):
    values[t] = rewards[t] + gamma*values[t+1]
# Plot values
plt.bar(times,values,color='r')
plt.title("Values",fontsize=15)
plt.xlabel("Time",fontsize=15)
plt.show()
```



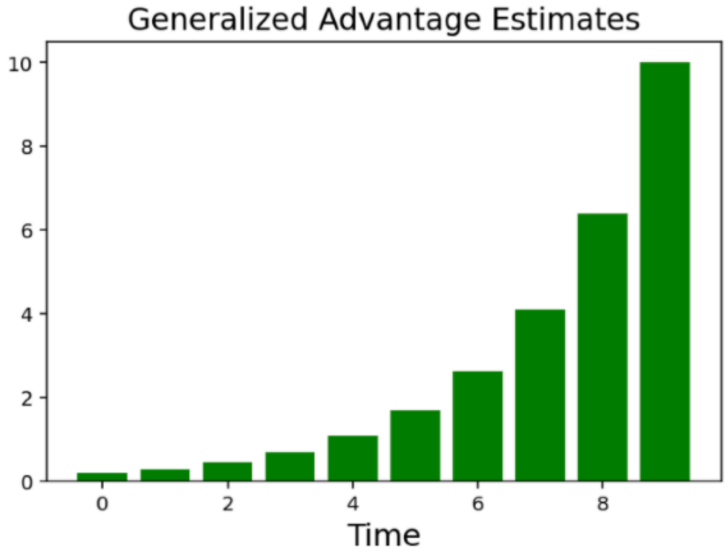
Advantage Function Estimation

Generalized Advantage Estimates

$$\delta_t^V = r_t + \gamma V(s_{t+1}) - V(s_t) \text{ - (9)}$$

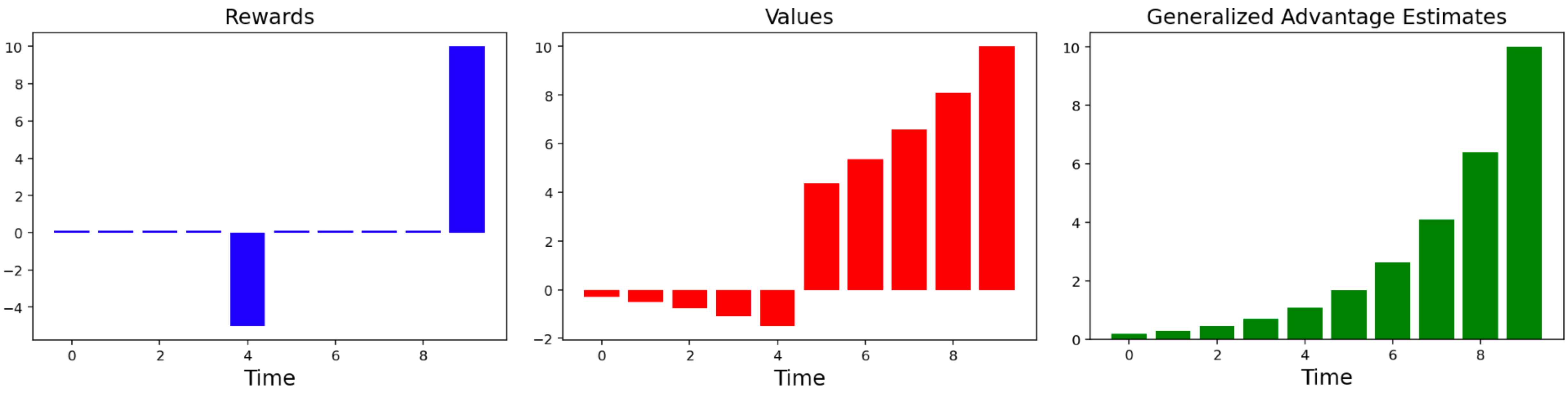
$$\hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)} = \sum_{l=0}^{\infty}(\gamma\lambda)^l \delta_{t+l}^V \text{ - (16)}$$

```
gaes = np.zeros(L); gaes[L-1] = rewards[L-1]
for t in reversed(range(L-1)):
    delta = rewards[t] + (gamma*values[t+1]) - values[t]
    gaes[t] = delta + (gamma*lamda*gaes[t+1])
# Plot GAEs
plt.bar(times,gaes,color='g')
plt.title("Generalized Advantage Estimates",fontsize=15)
plt.xlabel("Time",fontsize=15)
plt.show()
```



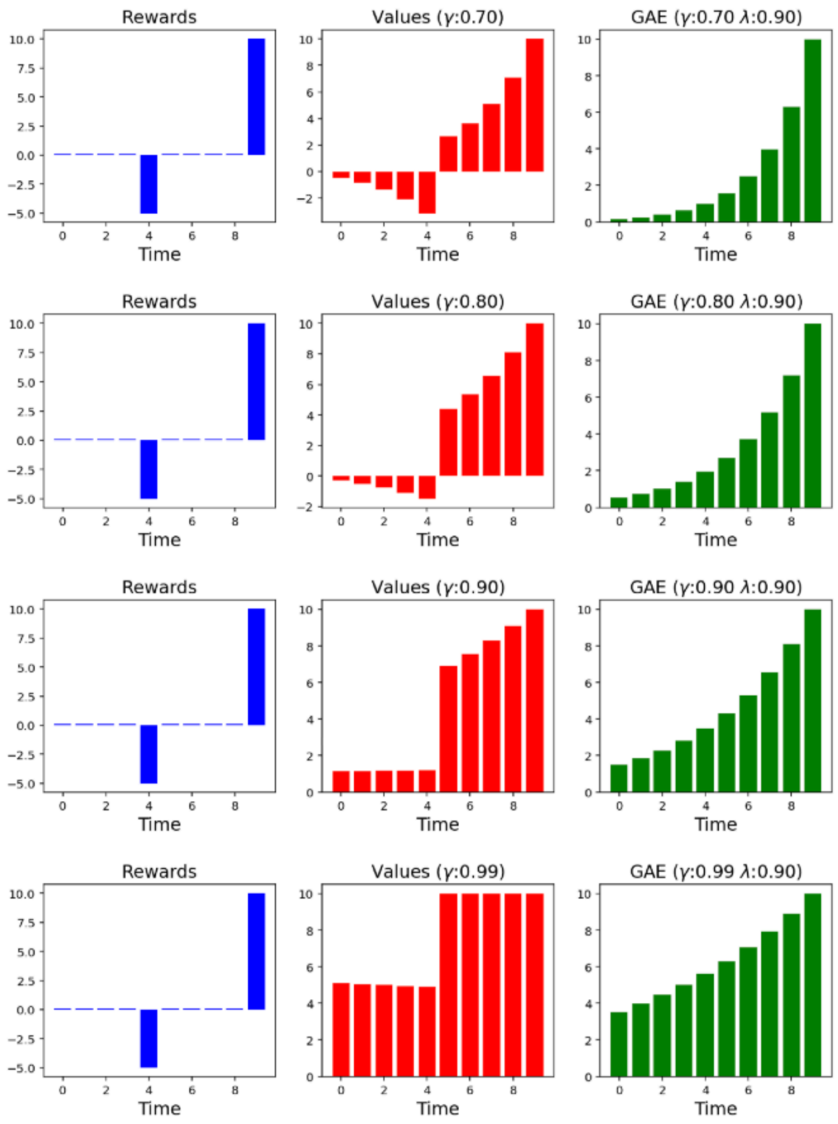
Advantage Function Estimation

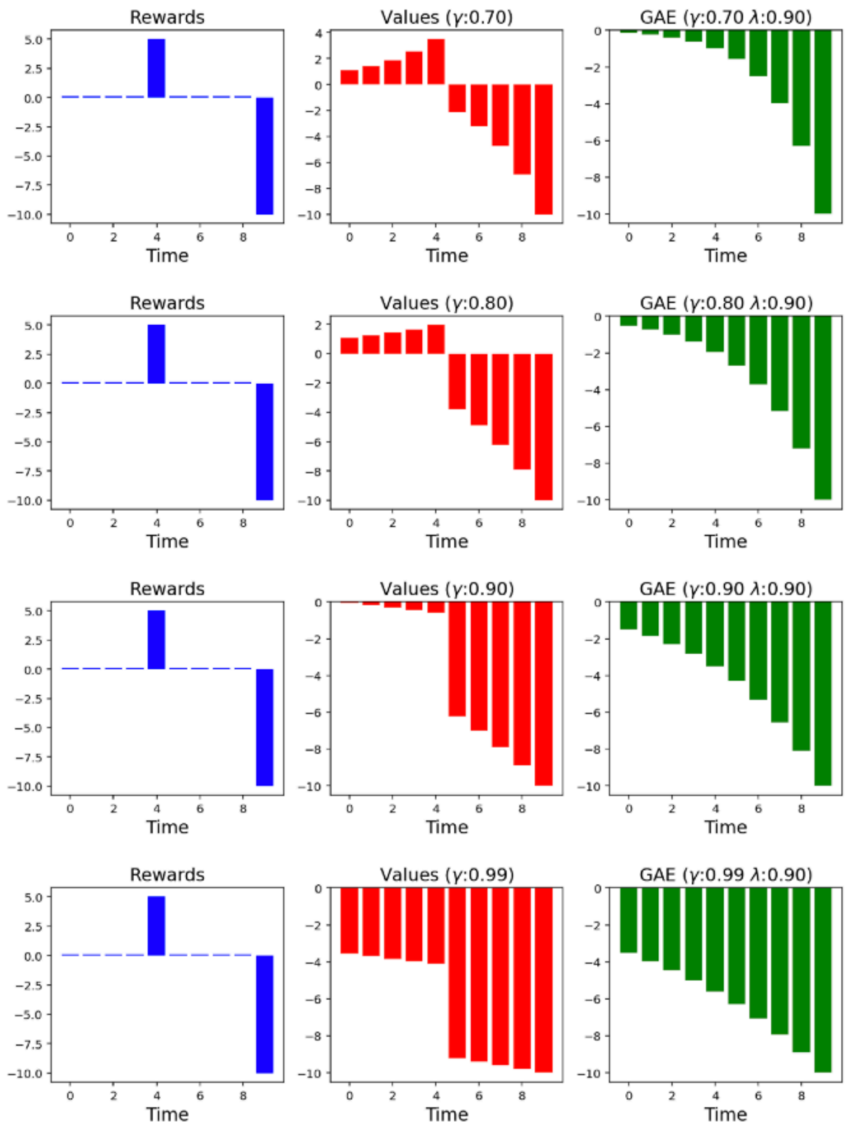
$$\hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)} = (1-\lambda)\left(\hat{A}_t^{(1)} + \lambda\hat{A}_t^{(2)} + \lambda^2\hat{A}_t^{(3)} + \cdots\right) = \sum_{t=0}^{\infty}(\gamma\lambda)^l \delta_{t+1}^V$$



https://gist.github.com/sjchoi86/38c7a378cfa482a1cde5630e5dde937e

Advantage Function Estimation





ROBOT INTELLIGENCE LAB

Soft Actor-Critic (SAC)

"Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," 2018

Maximum Entropy RL

ROBOT INTELLIGENCE LAB

- Standard RL objective

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[r(s_t, a_t)\right]$$

- Maximum Entropy RL objective

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[r(s_t, a_t) + \alpha\mathcal{H}\left(\pi(\cdot \mid s_t)\right)\right]$$

Maximum Entropy RL

- Maximum Entropy RL objective

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[r(s_t,a_t) + \alpha\mathcal{H}\left(\pi(\cdot\,|\,s_t)\right)\right]$$

- **Policy evaluation step**
    - The Bellman backup operator for Max-Ent RL is:

$$T^\pi Q(s_t,a_t) \triangleq r(s_t,a_t) + \gamma\mathbb{E}_{s_{t+1}}\left[V(s_{t+1})\right]$$

where $V(s_{t+1}) = \mathbb{E}_{a_t\sim\pi}\left[Q(s_t,a_t) - \log\pi(a_t\,|\,s_t)\right].$

- **Policy improvement step**

$$\pi_{new} = \arg\min_{\pi'} D_{KL}\left(\pi'(\cdot\,|\,s_t)\,\Big\|\,\frac{\exp(Q^{\pi_{old}}(s_t,\cdot))}{Z^{\pi_{old}}(s_t)}\right)$$

Soft Actor-Critic

- SAC learns three functions: $V_\psi(s)$, $Q_\theta(s,a)$, and $\pi_\phi(a \mid s)$.
- For learning $V_\psi(s)$:

$$J_V(\psi) = \mathbb{E}_{s_t \sim \mathcal{D}}\left[\frac{1}{2}\left(V_\psi(s_t) - \mathbb{E}_{a_t \sim \pi_\phi}\left[Q_\theta(s_t, a_t) - \log \pi_\phi(a_t \mid s_t)\right]\right)^2\right]$$

where actions are being sampled from the current policy $\pi_\phi(a \mid s)$ not from the replay.

- For learning $Q_\theta(s,a)$:

$$J_Q(\theta) = \mathbb{E}_{(s_t,a_t) \sim \mathcal{D}}\left[\frac{1}{2}\left(Q_\theta(s_t, a_t) - \hat{Q}(s_t, a_t)\right)^2\right] \text{ where } \hat{Q}(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}}\left[V_\psi(s_{t+1})\right]$$

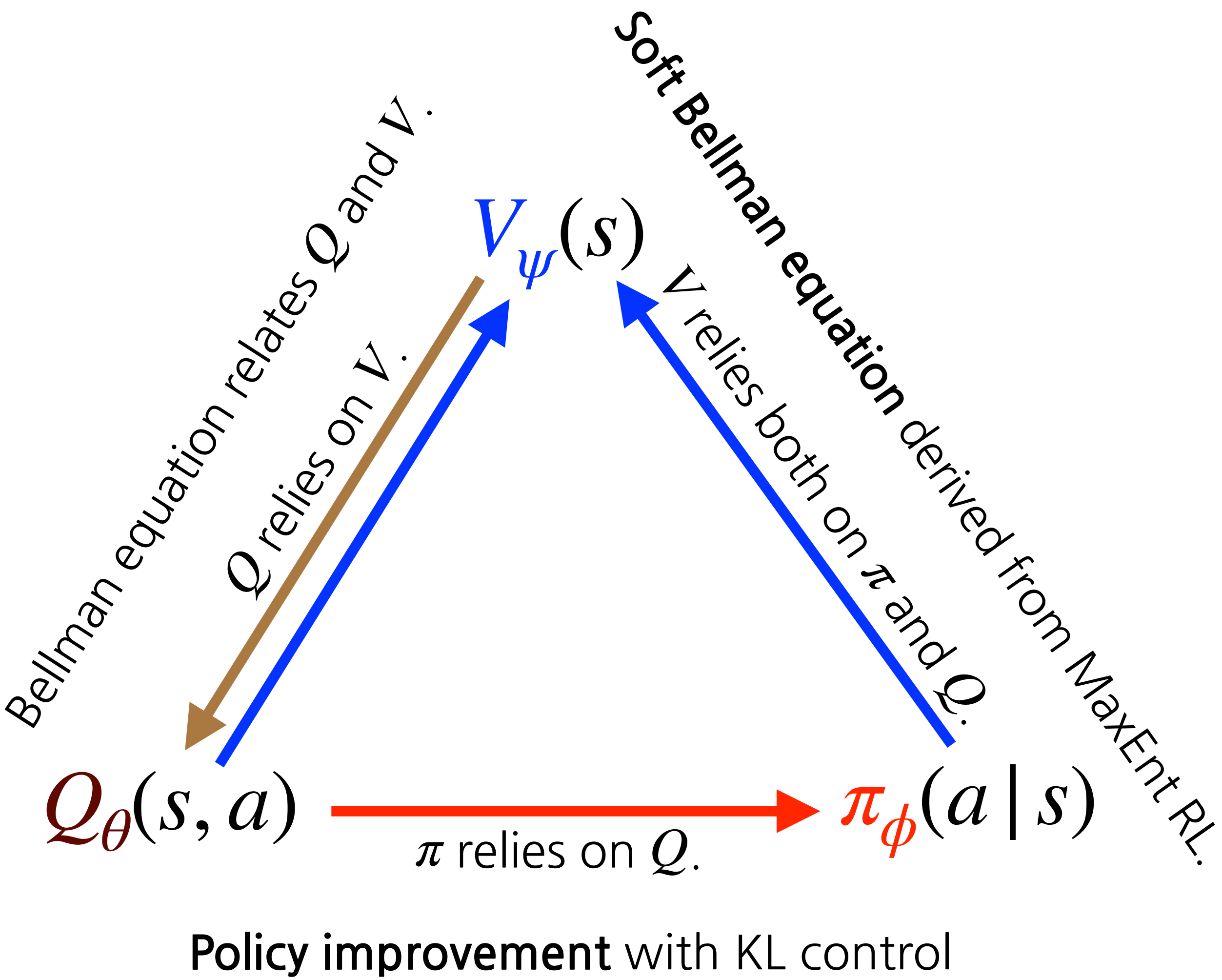
- For learning $\pi_\phi(a \mid s)$:

$$J_\pi(\phi) = \mathbb{E}_{s_t \sim \mathcal{D}}\left[D_{KL}\left(\pi_\phi(\cdot \mid s_t) \middle\| \frac{\exp\left(Q_\theta(s_t, \cdot)\right)}{Z_\theta(s_t)}\right)\right]$$

If we reparameterize the stochastic policy $a_t = f_\phi(\epsilon_t; s_t)$ where ϵ_t is sampled from some distribution,

$$J_\pi(\phi) = \mathbb{E}_{s_t \sim D, \epsilon_t \sim \mathcal{N}}\left[\log \pi_\phi\left(f_\phi(\epsilon_t; s_t) \mid s_t\right) - Q_\theta\left(s_t, f_\phi(\epsilon_t; s_t)\right)\right]$$

Soft Actor-Critic



Summary

- Policy Gradient Theorem
  - Optimize the policy directly via $\nabla_\theta \eta(\pi_\theta) \approx \nabla_\theta \log \pi_\theta(a_t | s_t) Q_{\pi_\theta}(s_t, a_t)$
- Trust Region Policy Optimization (**TRPO**)
  - From policy improvements using minorization maximization to a trust-region method.
- Proximal Policy Optimization (**PPO**)
  - Approximate TRPO with policy ratio clipping and adaptive KL weights.
- Generalized Advantage Estimation (**GAE**)
  - More robust than the value estimate, similar to TD(λ).
- Soft Actor-Critic (**SAC**)
  - Entropy-regularized RL with an actor-critic method.

Thank You

ROBOT INTELLIGENCE LAB